(54) **Dialogue system**

(57) Interactive dialogue system comprising a speech recogniser (11) for analysing a user's utterances and a speech synthesiser for transmitting messages to the user. The system includes a dialogue controller including an intelligent knowledge base (IKBS) (15) comprising frame based knowledge representation having a hierarchy of frames containing information about the dialogue. Each frame has slots having one or more values denoting atomic values, references to sub-frames, or procedures. The dialogue controller also includes a linguistic processor (13) which converts a word string supplied by the recogniser into the high level semantic representation of the IKBS and uses high level data from the IKBS to assist in recognising the next statement spoken by the user. The system may obtain information to answer a user enquiry from a database 17, or direct a computer to carry out an instruction or an appliance to alter its function.
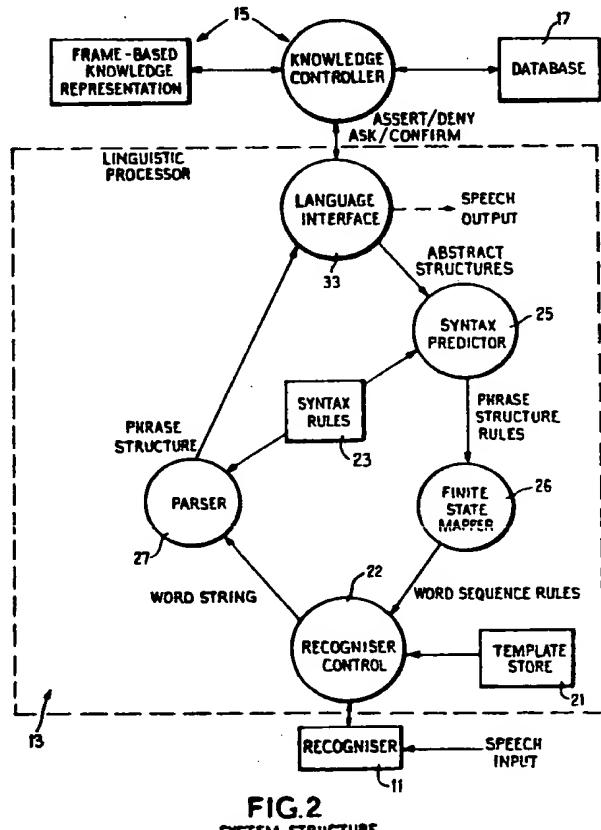
FIG.2
SYSTEM STRUCTURE
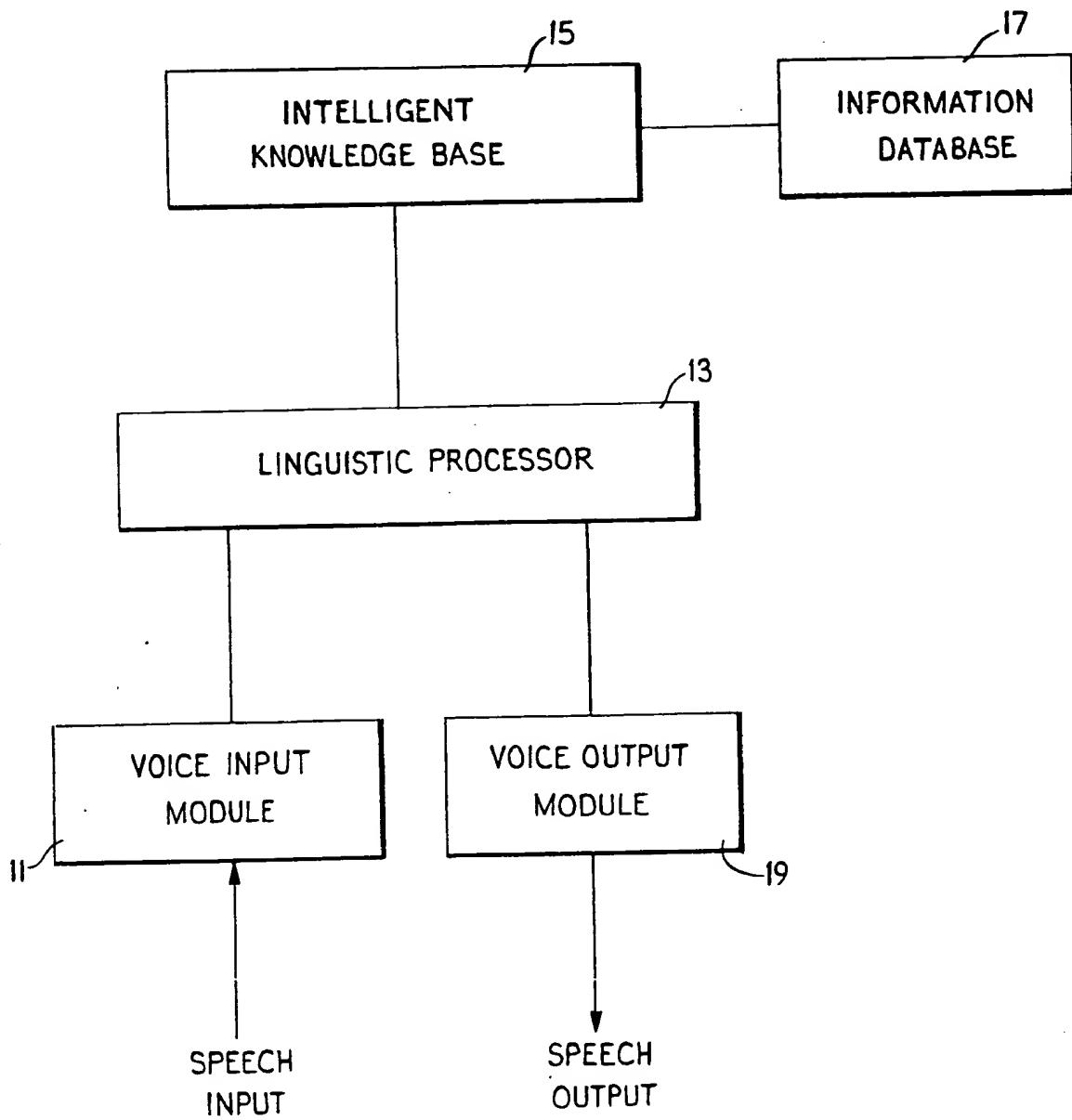
FIG.1

OVERALL STRUCTURE OF VOICE INFORMATION SYSTEM

2165969

FIG.2
SYSTEM STRUCTURE

DIALOGUE 1

| WHAT & TRAVEL |
| TRAVEL |
| COST |

TRAVEL 1

| LEAVE |
| ARRIVE |
| TRAIN |

EVENT 1

| LOCATION |
| Manchester |
| TIME |

INTERVAL 1

| SOONEST |
| ABOUT |
| LATEST |

MOMENT 1

| HOUR 9 |
| MIN 0 |

EVENT 2

| LOCATION |
| London |
| TIME |

INTERVAL 2

| SOONEST |
| ABOUT |
| LATEST |

MOMENT 2

| HOUR 12 |
| MIN 0 |

TRAIN 1

| FROM Picadilly |
| TO   Euston |
| START |
| FINISH |

MOMENT 3

| HOUR 9 |
| MIN 16 |

MOMENT 4

| HOUR 11 |
| MIN 58 |
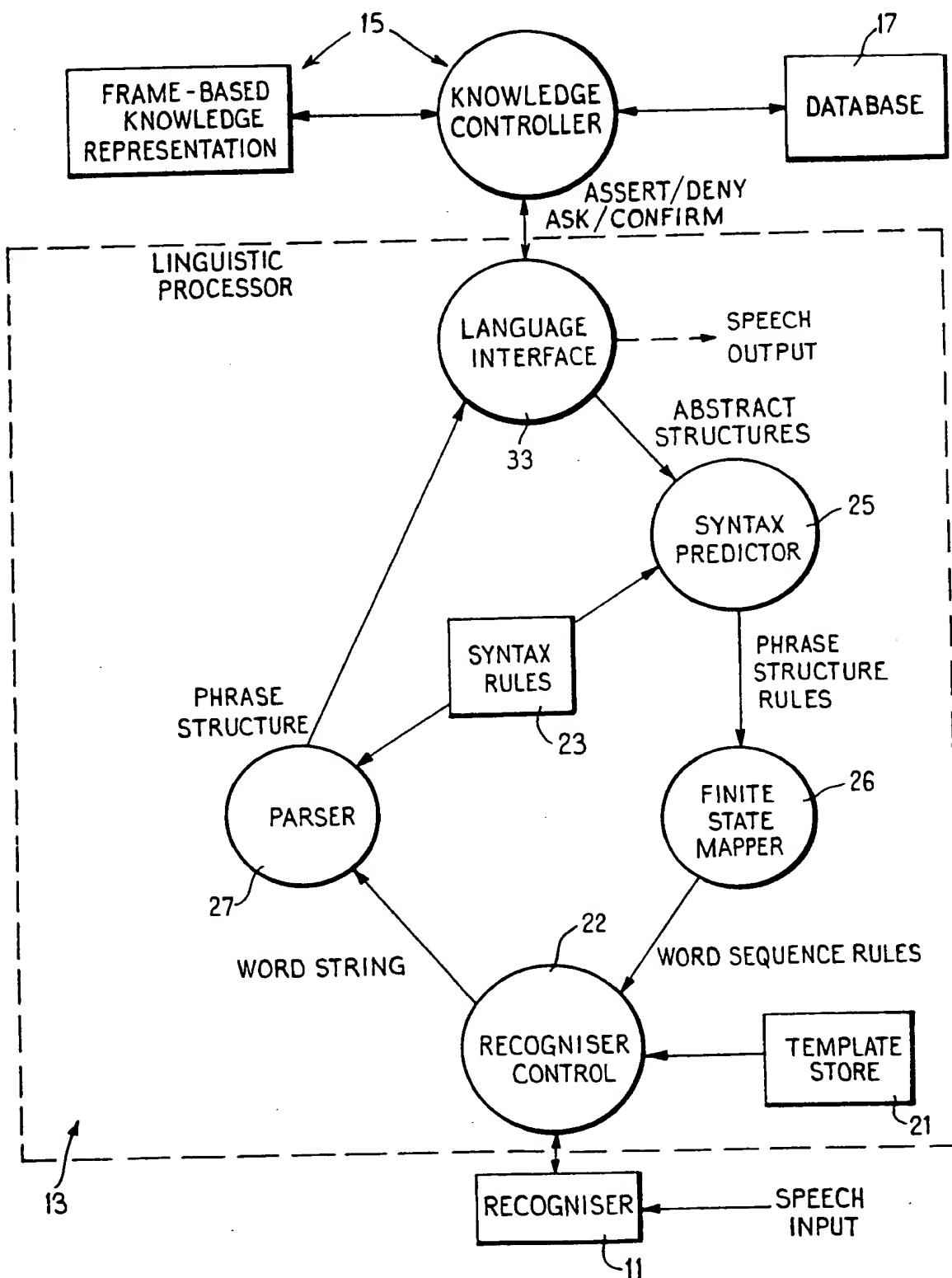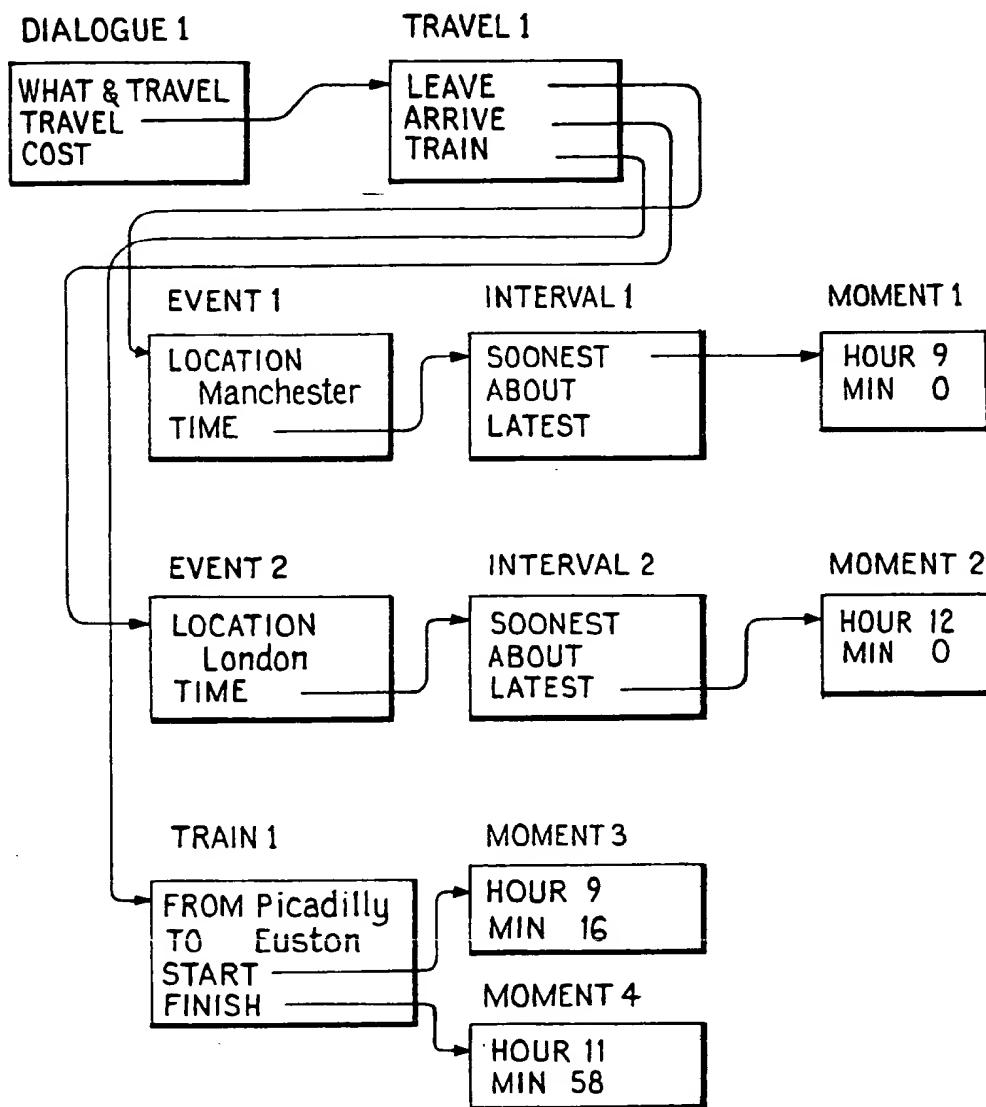
FIG.3

INSTANTIATED FRAME STRUCTURE

£ASSERT
REQUIRE   £TRAVEL
                QUAL.
        WANTVERB
                PREP   £LOCATION
I   WANT   TO GO   TO   LONDON

QUAL
£LOCATION
LEEDS

QUAL
PREP   £LOCATION
TO   BRISTOL

QUAL
        £LOCATION
NOT   BRIGHTON

£ASSERT
REQUIRE   £TRAVEL
                        QUAL
                QUAL   £TIME
        WANTVERB   PREP £LOCATION PREP £NUMBER £NUMBER
I   WANT   TO ARRIVE   IN   LONDON   BY   TEN   THIRTY   AM

FIG.4

questions not yet asked, and to answer indirectly. Thus the syntax predictor must select phrase structure rules for a range of possible responses. As the IKBS may have asked for confirmation, the caller may w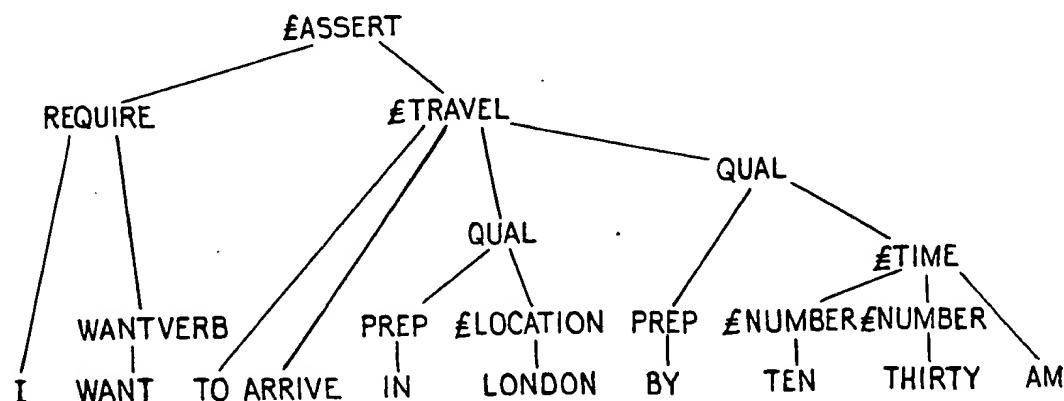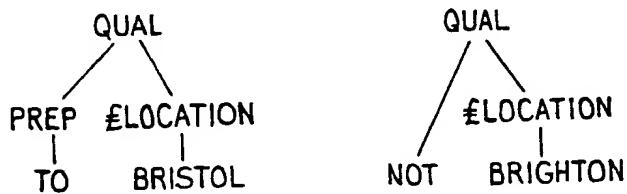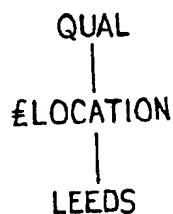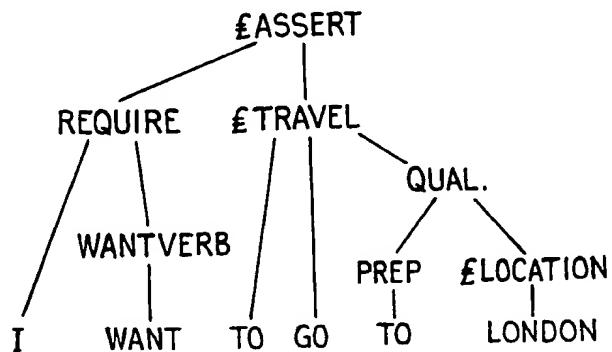ish to make a denial and appropriate syntax rules must be available. If there is a large number of different possible responses, the recogniser may be unable to handle the
5 number of possibilities and it may be necessary to exclude some possibilities; and if recognition fails, repeat the question in a different way or use a different set of syntactic predictions. The parser 27 is not required to perform a complete parse of the input, but to produce as complete a parse as possible. The resultant high level semantic representation is passed to the language interface 33 and to IKBS 15 in the normal way, and the interface 33 and IKBS 15 cooperate in
10 attempting to infer the likely meaning. In making an inference, the system uses the current focus (ie. the questions currently being asked and responded to) and the current state of the IKBS. Incorrect inferences can be corrected during a subsequent "confirm" request. The caller may be asked to repeat the answer.
   Once the caller's query has been established, the next stage is for the IKBS 15 to consult train
15 timetable database 17. The input to and output from the database is in frame format. Train time data is stored externally in a text file. On start-up of the system, the database 17 reads the text file and stores it in a structure adapted for fast access. The data comprises train times and routes, arranged in a manner similar to that of a normal timetable. The information supplied by the database to the IKBS may be the details of a train or trains, or the indication that no
20 suitable train can be found. The information is relayed to the caller in the same way as questions or confirm requests during earlier stages of the dialogue.
   The "Logos" speech recogniser used in the above example is a connected word recogniser. If desired, an isolated word recogniser could be used, but the dialogue possible would be more restricted. In some circumstances this may be satisfactory, and the system would need less
25 complex linguistic processing. For example, inexperienced users may tend to use complex sentence structures, resulting in poor recognition scores from a connected word recogniser. If restricted to one-word replies, a more satisfactory dialogue may result. Isolated word recognisers may be used where a relatively cheap system is required. In practice, currently available speech recognisers require predicted response information, eg. in the form of word sequence
30 rules from a syntax predictor, in order to predict the word order of possible responses and reduce the number of recognition possibilities. The form and operation of the linguistic processor in any embodiment of the invention will depend on the nature of the recogniser, the IKBS and the application of the system.
   Any suitable recogniser or speech synthesiser may be used in the system. In addition to
35 providing the parser with a word string, the recogniser may provide alternative word string(s) and/or confidence levels. As indicated above, the output may be via a speech synthesiser or a VDU.
   In the system described above, IKBS 15 sends data to language interface 33 including a question in textual form embedded as a string in a frame, to be passed to speech synthesiser
40 19 via a handler process. If desired, all data sent by the IKBS 15 to interface 33 could be in a high level semantic representation and a message generator could be provided between the interface and the speech synthesiser 19. Such a message generator would contain a set of syntax rules (similar to those in syntax predictor 25) for producing grammatical phrases and sentences to be transmitted to the caller via the speech synthesiser. A sophisticated speech
45 synthesiser would be able to make a comprehensible statement from such an input. A machine such as Prose 2000, however, may also benefit from additional information on speech production and word pronounciation, including details of appropriate stresses and intonations, pauses etc. Processors to provide the necessary rules for this would need to be included in the message generator.
50   In the above example, a train timetable service is provided to telephone customers who, on dialling the number of the service, are asked a series of questions by a speech synthesiser and are given appropriate timetable information. Instead of a train timetable database, the dialogue controller may be interfaced with other kinds of database, a computer (eg. a bank's computer), a domestic or industrial appliance, an office's central heating system etc. The dialogue controller
55 acts as an intermediary between a user and the database, computer or appliance and obtains information from it, supplies information to it, and controls it in accordance with the user's request or instruction. The user may, but need not, access the system by telephone.

   CLAIMS
60   1. Interactive dialogue system, comprising a speech recogniser arranged to analyse a user's utterances, transmission means for transmitting messages to the user and a dialogue controller including an intelligent knowledge base comprising frame based knowledge representaton having a hierarchy of frames containing information about the dialogue, wherein the dialogue controller is arranged to accept and interpret output relating to a user's utterance from the speech
65 recogniser and to supply data to the transmission means for the transmission of a message to

the user.

2.   Interactive dialogue system as claimed in Claim 1 for responding to a user request, wherein the dialogue controller is arranged to transmit one or more directions relating to the request to an auxiliary device.

3.   Interactive dialogue system as claimed in Claim 2, including said auxiliary device, wherein the device comprises a data store containing data necessary for responding to the user's request and the dialogue controller is arranged to supply the response data to the transmission means.

4.   Interactive dialogue system as claimed in Claim 2, adapted for a user request to operate or modify the operation of the device, wherein the direction or directions transmitted by the dialogue controller carry out the required operation or modification.

5.   Interactive dialogue system as claimed in any preceding claim, wherein the frame based knowledge representation is structure so as to determine both the conduct of the dialogue and the operation of the intelligent knowledge base.

6.   Interactive dialogue system as claimed in Claim 5, wherein the frames comprise slots, and at least one of the frames includes slots denoting procedures.

7.   Interactive dialogue system as claimed in Claim 6, wherein said intelligent knowledge base is arranged to accept and/or generate a high level semantic representation of data.

8.   Interactive dialogue system as claimed in Claim 7, wherein the dialogue controller includes linguistic processing means arranged for converting output from the speech recogniser to said high level semantic representation.

9.   Interactive dialogue system as claimed in Claim 8, wherein the intelligent knowledge base is arranged to send predicted response information to the speech recogniser to constrain the speech recogniser to recognise only a limited set of utterances and/or series of utterances.

10.   Interactive dialogue system as claimed in Claim 9, wherein the predicted response information is initially in said high level semantic representation and said linguistic processing means is adapted to convert said information to a lower level semantic representation prior to input to the speech recogniser.

11.   Interactive dialogue system as claimed in any one of claims 8 to 10, wherein the linguistic processing means is arranged for converting data in said high level semantic presentation to a lower level semantic and syntactic representation for output to the transmission means.

12.   Interactive dialogue system as claimed in any one of Claim 8 to 11, wherein the linguistic processing means is adapted to receive alternative sets of data of different confidence levels from the speech recogniser corresponding to a single user utterance, and the process said sets of data for use by the intelligent knowledge base.

13.   Interactive dialogue system as claimed in any preceding claim, wherein the transmission means comprises a speech synthesiser.

14.   Interactive dialogue system as claimed in any preceding claim, wherein the speech recogniser is a connected word recogniser.

15.   Interactive dialogue system as claimed in of Claims 1 to 13, wherein the speech recogniser is an isolated word recogniser.

16.   A method of conducting a dialogue with a user to establish a request, comprising supplying voice signals derived from a user utterance to a speech recogniser, supplying output from the speech recogniser relating to the voice signals to a dialogue controller including an intelligent knowledge base comprising frame based knowledge representation having a hierarchy of frames containing information about the dialogue; interpreting said output; transmitting a message to the user; and repeatedly interpreting output relating to user voice signals from the speech recogniser and transmitting messages to the user to establish the user request, and responding to that request.

17.   A method as claimed in Claim 16, including consulting a database to obtain information necessary for responding to the user request and transmitting said information to the user.

18.   A method as claimed in Claim 16, including establishing a user request to operate or modify an auxiliary device and transmitting instructions to the auxiliary device to carry out the required operation or modification.

19.   A method as claimed in any one of Claims 16 to 18, including converting the output of the speech recogniser to high level semantic representation and converting data in said high level representation to a lower level semantic and syntactic representation for transmitting a message to the user.

20.   An interactive dialogue system substantially as hereinbefore described, with reference to the accompanying drawings.

21.   A method of conducting a dialogue with a user substantially as hereinbefore described, with reference to the accompanying drawings.

synthesiser 19 via a simple handler process (not shown). The ask request is then spoken by
synthesiser 19 to the caller.

In addition to data being sent via interface 33 to the synthesiser 19, the IKBS also sends data
to aid the speech recogniser and linguistic processor in processing the caller's response. The
5 data, in the form of a frame structure, from the IKBS 15 is encoded into an abstract (ie. non-   5
linguistic) form and passed to syntax predictor 25. For example, the coded form of the question
"Where to you want to go to?" may be:
    £TRAVEL (£ARRIVE(£LOCATION?)))
Syntax predictor 25 holds a set of syntax rules describing all possible utterances to be
10 recognised by the recogniser 11 in response to the question. All rules which could potentially be   10
used by the caller in stating the destination are extracted and passed to finite state mapper 26.
Some of the rule definitions supplied by store 23 may be preceded by a character, shown as £
in the examples given below, indicating that phrases descended from this rule carry significant
semantic information; these may correspond to slot names in the IKBS frames.
15 The rules from store 23 are context free phrase structure rules. Some of these rules are given   15
below:

```
SENTENCE   → £ASSERT/£DENY/£QUERY/£YES/£NO
£ASSERT    → REQUIRE £TRAVEL
20 £DENY      → NOTREQ £TRAVEL                                                                  20
REQUIRE    → I WANTVERB
NOTREQ     → I DONT WANTVERB
WANTVERB   → WANT/WISH/NEED
£TRAVEL    → TO (LEAVE/ARRIVE/GO) {QUAL}
25 QUAL      → [NOT] [PREP] (£LOCATION/£TIME/£MODE)                                             25
£LOCATION  → LONDON/LEEDS/MANCHESTER/ ...
£TIME      → £{NUMBER} [AM/PM]
£MODE      → (FIRST/SECOND) CLASS/PULLMAN
£NUMBER    → ONE/TWO/ .../THIRTY/FORTY/FIFTY
30 PREP       → FROM/AT/TO/BY/ABOUT/BEFORE/AFTER/IN                                             30
```

where

```
[ ]  =option
35 , ,  =zero or more repetitions                                                              35
/    =alternatives
( )  =factor
```

Examples of phrase structures generated by these rules are shown in Fig. 4.
40 The selected syntax rules are then passed from syntax predictor 25 to finite state mapper 26,   40
which builds a finite state network representing the possible word sequences which recogniser
11 should look for. This network, together with the necessary word templates from store 21 is
passed to the recogniser 11 and the caller's speech input is processed.
The 'Logos' recogniser 11 is designed to find the sequence of templates which give the best
45 match with the speech input. It is necessary, in order for the matching to be feasible relatively   45
quickly, for the search to be constrained by the finite state network. For example, a simplified
version of the rules for the above query "Where do you want to go to?" may be:

```
togo      =to go;
50 toplace   =to place;                                                                         50
fromplace =from place;
query     =togo [fromplace/toplace],
```

where [ ] denotes an option and / denotes an alternative. The recognition network for 'query'
55 would be the following word sequence matrix: 55

	isil	to	from	go	place	fsil
isil	0	1	0	0	0	0
to	0	0	0	1	1	0
from	0	0	0	0	1	0
go	0	1	1	0	0	0
place	0	0	0	0	0	1
fsil	0	0	0	0	0	0

In this matrix, a 1 indicates that the word labelling its column immediately follows the word labelling its row (isil and fsil denote initial and final silence, respectively).

The output from recogniser 11 is a compacted data sequence representing a word string, with, possibly, mis-identified words and omissions. The string is passed to parser 27, which uses the set of syntax rules selected by predictor 25 and used by finite state mapper 26 and recogniser 11. As much of the input as possible is parsed into a phrase structure tree by the parser. Note that the parser will parse whichever is available, whether this is a single word (eg. LEEDS in Fig. 4), a phrase (TO BRISTOL, NOT BRIGHTON in Fig. 4), or a complete sentence. Any missing words may have been unrecognisable or not spoken. The parser scans the word string from left to right attempting to find a substring that matches the right hand side of a syntax rule. Once a match is found, the word or words are replaced by the right hand side of the rule and the process is repeated on the reduced string until no further matches can be located. There is often more than one possible match at any stage and sometimes the string may not be reduced as far as possible on the first attempt. If the word string is not fully replaced, matches are undone and different matches tried to find the most complete reduction possible.

The parsed word string is then further processed by parser 27 and passed to linguistic interface 33. Examples of some rules applied by parser 27 are as follows:

QUAL (PREP([BY/BEFORE]) £TIME(x))→QUAL(£TIME(£LATEST(x)))
QUAL (PREP([AT/ABOUT]) £TIME(x))→QUAL(£TIME(£ABOUT(x)))
QUAL (PREP(AFTER) £TIME (x)))→QUAL(£TIME(£SOONEST(x)))
QUAL(PREP(TO) £LOCATION(x)))→£ARRIVE (£LOCATION(x))
QUAL (PREP(FROM) £LOCATION(x))→£LEAVE (£LOCATION(x))
£TRAVEL(TO ARRIVE QUAL(x) y)→£TRAVEL (£ARRIVE(x) y)
£TRAVEL (TO LEAVE QUAL(x) y)→£TRAVEL (£LEAVE(x) y)

where x,y denote variables (eg. a sequence of zero or more nodes in Fig. 4) [ ] denotes an optional node

The phrase structures are reproduced by a nested list notation. For example, the phrase structure for "I want to go to London", shown in Fig. 4, would take the following form:

£ASSERT (REQUIRE(I WANTVERB(WANT)) £TRAVEL (TO GO QUAL (PREP(TO) £LOCATION (LONDON))))

Each rule is matched against the phrase structure and, for each match, the matched segment is replaced by the right hand side of the rule. Each rule is applied to the whole phrase structure before moving to the next rule.

The above rule would be transformed as follows:

£ASSERT (REQUIRE(I WANTVERB(WANT)) £TRAVEL (TO GO £ARRIVE (£LOCATION (LONDON))))

Once all rules have been applied, all non-terminal nodes not preceded by the symbol £ are deleted giving in this case:

£ASSERT (£TRAVEL(£ARRIVE(£LOCATION(LONDON))))

This abstract phrase structure is now in the form of a high level semantic representation which can be used directly by the IKBS 15. The sequence £TRAVEL(£ARRIVE(£LOCATION( ))) is used to map a path through the frame structure of the IKBS to assign the value LONDON.

A single transaction cycle, in which the caller is asked a question and data from a statement from the caller is supplied to the IKBS, has been described above.

In order to ask the caller another question (following on from the answer which has been received, that the caller wishes to travel to London), an appropriate frame structure is sent from the IKBS 15 to language interface 33 of linguistic processor 13. The caller is asked another question, the response processed and data passed to IKBS 15. The above procedure is repeated for further question and answer cycles. Cycles are performed until all the information necessary to answer the enquiry has been accumulated.

The above description is simplified as in practice the caller must be permitted to answer

default value so the associated "if-needed" procedure is not triggered. However, the slot also
has an "if-added" procedure and this is triggered to confirm the departure location.

5. The star on the TIME slot of EVENT causes an instance of the INTERNAL frame to be
instantiated. Instantiation of a needed frame value is automatic unless some special action is
required such as the inclusion of a default value as in the case of EVENT above, and an "if
needed" procedure "INSTANTIATE (TIME)" is triggered automatically. This creates an instance
of the INTERNAL frame.

6. The star on the ABOUT slot of INTERVAL causes the associated 'ask (ABOUT)' procedure
to trigger. When the user answers specifying a time, the "if added" procedure triggers and
computes values for the SOONEST and/or LATEST slots. ABOUT is next instantiated, and
instances of the MOMENT frame are created, and values for the HOUR and MINUTE slots written
in.

7. The instantiation of the LEAVE slot in TRAVEL is now complete. Next, the ARRIVE slot is
instantiated and a similar sequence of events occurs to establish values for the arrival location
and earliest and/or latest possible arrival times.

8. When all of the arguments of the original 'lookup' procedure in the TRAIN slot of the
TRAVEL frame are known, the required database access is made. Asserting a value for the
TRAIN slot then triggers the 'tell-user-about' procedure and the requested information in output.

Fig. 3 shows diagrammatically the final frame structure which would be instantiated during the
above sequence of events assuming that the dialogue proceeded as follows:

Q1 What information do you require?
A1 Train times.
Q2 I assume that you wish to travel from Manchester. When do you want to leave?
A2 By 9a.m.
Q3 Where do you want to go to?
A3 London.
Q4 When do you want to arrive in London?
A4 By lunchtime
R The 9.16 from Picadilly Station arrives at Euston Station at 11.58a.m.

It will be appreciated that the above example is highly simplified. In practice, the DIALOGUE
frame would have many more slots which would be updated automatically by the "ask" and
"confirm" procedures.

The association of "ask" procedures with the data in the IKBS ensures that questions are
asked only for data which is actually needed. As indicated in paragraph 5 above of the frame
instantiation procedure, default values may be placed in frames during instantiation; alternatively
they could be included in the generic frames from the start.

The current focus of attention is represented by the slot which is active at any instant.
Currently active slots are updated automatically by the "ask" and "confirm" procedures. If
unsolicited information is given, which does not match any value in the currently active slot, a
search can be made of adjacent slots and further frames may be instantiated. Once a slot which
matches the data has been found, the value is filled in. The IKBS can then continue with the part
of the dialogue controlled by the new slot or frame to avoid a non-sequitur in the dialogue, and
subsequently return to the original slot. The "if needed" procedures of the completed slots will
not be triggered again later in the dialogue as their values are already present; this prevents the
system asking questions to which answers have already been given.

Programming languages may be developed to facilitate the programming of frame based
systems. The language which has been developed for the system described above is
named "UFL" and is based on conventional software routines modified so as to be able to define
frames constituting the entire program ie. both the structure of the data and its execution. This
particular language may be installed on most machines equipped with an ISO standard Pascal
Compiler. Note that the examples of frames given above do not adopt the precise syntax of
UFL; it is not necessary to know the details of this syntax in order to implement a system
according to the invention. The examples given below illustrate, using UFL, the principles in-
volved in integrating procedures into a frame system.

Using UFL, frames can be defined using a simple textual notation as illustrated below. The
algorithms used are not application dependent, so that the dialogue system could be modified
for a different application by re-writing the specifications of the generic frames. Re-defining
frames causes changes to propagate through the whole frame structure. This flexibility is pos-
sible because the software procedures are embedded within the frame structure in an analogous
fashion to the way in which data is incorporated in known frame structures. The implementation
of procedures in existing frame systems is usually heavily dependent on the special purpose
language compiler used.

A UFL program consists of a set of frame definitions as shown in the following example of a
frame called "person" (which is unrelated to the train timetable application discussed above):

```
person
(ako: standard,
*name: person-name,
*age: int
)
```

The slot definitions are separated by commas and enclosed in parentheses. Each slot definition contains the name of the slot and its value. A colon after the name of a slot indicates that the value of the slot is to be found in a sub-frame; the name of the relevant sub-frame is given in the value field. The above frame contains three slots called 'ako', 'name', and 'age'. The value of the "ako" slot can be found in the sub-frame "standard".

The 'ako' (a kind of) slot is of particular importance. Normally each frame includes an 'ako' slot which defines the location of the frame within the hierarchy of frames. By use of the 'ako' slot in conjunction with the inheritance mechanism, it is possible to have a single generic frame at the top of the hierarchy called 'standard' which comprises all the procedures. The 'standard' frame includes a large number of slots denoting procedure values, eg 'inst' (instantiate), 'read' and 'write'. Procedures do not need to be included in sub-frames, as frames lower down in the hierarchy automatically refer to 'standard' or other frames via the 'ako' slot. By modifying the 'standard' frame the characteristics of the whole or part of the performance of the system can be altered.

When an attempt is made to instantiate the 'person' frame shown above, the IKBS searches for an 'inst' (instantiate) procedure in the frame. On failure to find such a procedure, the system uses the 'ako' slot which causes the system to search back through the inheritance hierarchy of the 'person' frame until the 'standard' frame containing procedures is reached.

Instantiation of the 'person' frame begins by instantiating the first of its 'needed' slots (those marked by stars (*), in this case the 'name' slot). If any of the needed slots referred to sub-frames which also contained starred slots, then their 'needed' slots would be instantiated. This continues through the heirarchy until a slot containing an atomic value or procedure is reached. When a value is to be assigned, an atomic value is written into the slot and when a procedure is encountered, it is executed. When a value is to be assigned to the slot of a frame, the value is passed to the procedure in a 'write' slot. Similarly, when a frame is to be read (eg. in order to transmit its value to the user) the procedure in a 'read' slot is executed, and all other operations are carried out by executing procedures. This provides great flexibility as it permits system procedures to be written and frames to be defined using those procedures.

In the above example, the values of the 'name' and 'age' slots, 'person-name' and 'int' (integer) are the names of other frames.

The dialogue controller also includes the linguistic processor 13, which is shown in more detail in Fig. 2. Linguistic processor 13 interfaces the IKBS 15 with the speech recogniser 11 and speech synthesiser 19. The IKBS 15, as indicated above, handles abstract data in a high level semantic representation (HLSR), and the processor 13 is used to convert speech input from recogniser 11 to the HLSR and the output from the IKBS to words, sentences or phrases to assist in understanding the next speech input from recogniser 11.

Linguistic processor 13 comprises several software processes, all defined in a form compatible with ISO PASCAL in order that the dialogue controller (IKBS 15 and processor 13) can communicate with an external computer having a suitable software compiler. Included in processor 13 are data stores 21, 23. Template store 21 is used to store all templates representing words in the vocubulary of the system, ready for loading into the 'Logos' recogniser 11 by recogniser control 22 for matching with comparable signals derived from spoken words delivered to the recogniser. Syntax rules store 23 comprises definitions of the set of syntax rules used by the system, and is accessed by syntax predictor 25, which sends phrase structure rules to finite state mapper 26 and parser 27 as indicated below. Output from parser 27 is supplied to language interface 33, which interfaces with the IKBS 15. Parser 27 also uses information generated by syntax predictor 25 to make parsing more efficient.

On start-up, recogniser control 22 of linguistic processor 13 initialises the 'Logos' recogniser 11, reads the template store 21 and loads relevent templates into the 'Logos' recogniser. The syntax rules are read by the parser 27 and syntax predictor 25.

The operation of the linguistic processor 13 is illustrated by an example. Suppose that a caller wishes to know the time of a train. In order to be able to give an answer, the system must determine the places of departure and arrival and the approximate departure and/or arrival time. This knowledge acquisition, as described above, is controlled by IKBS 15 which establishes questions to ask to caller to determine the enquiry, and accesses the database 17 of train timetable information prior to transmitting the answer to the enquiry. Suppose first question to be asked by the system is where the caller wishes to travel to. The data sent from IKBS 15 to language interface 33 includes an "ask" request in natural language, embedded as a text string in a frame; for example "Where do you want to go to?". This string is passed to speech

```
    *  FRAME:        & DIALOGUE

    *  WHAT          - - - - -      ? ask (WHAT)
 5                                                                              5
                                    = check (WHAT)

                                    + instantiate (WHAT)


      TRAVEL         & TRAVEL
10                                                                             10
      COST           & TICKET
```

The frame is named DIALOGUE and a symbol such as & indicators that it is a generic frame.
15 The frame is marked with a star and is at the top level of a hierarchy of frames. Initially, the   15
only frames present in the IKBS are generic frames; these represent static system knowledge.
During the course of a dialogue, new knowledge is acquired, and frames incorporating specific
values are established lower down the hierarchy. The process of creating and providing values
for specific instances of frames and slots is termed instantiation. The IKBS first creates an
20 instance of the starred top-level frame in the hierarchy (in this case the DIALOGUE frame).   20
    The DIALOGUE frame is divided into fields (columns) and slots (rows). The frame system is
arranged to acquire information from the user needed to complete any unfilled slots which are
preceded by a star (*). The processor starts with the top-level DIALOGUE frame, which as
indicated above, is itself marked with a star.
25 The first field (left hand column) of a slot (row) gives the name of the slot (WHAT, TRAVEL   25
etc.). The second field gives its value (if any) and a third field lists any associated procedures.
The procedures include one or more arguments such as the name of the slot WHAT in the
above example. The DIALOGUE frame has slots requiring values, but as it is a top-level generic
frame, values are not provided directly. The DIALOGUE frame needs values for WHAT, TRAVEL
30 and TICKET and these are provided by referring to frames lower down in the hierarchy. In the   30
WHAT slot of DIALOGUE, the value field is empty and the third field has associated procedures
to be carried out, leading to instantation of a WHAT sub-frame. TRAVEL and TICKET both have
values referring to generic sub-frames which are instantiated to provide values for the slots in
DIALOGUE. Thus, by following through the hierarchy the values required to complete the DIA-
35 LOGUE frame are obtained.   35
    As indicated above, procedures are triggered by events associated with their slot. The four
main triggers are "if needed" (?), "check"(=), "if added" (+) and "if inconsistent" (−). The
symbols shown here are used in the examples to indicate which procedures are associated with
a slot. "If needed" procedures trigger when an attempt is made to read a slot which has no
40 value (such as WHAT in the above example). "Check" procedures check whether the data which   40
are proposed to be written into a slot are reasonable (ie in the above example, whether TRAVEL
or TICKET is proposed as a value for the WHAT slot in response to the "if needed" procedure).
"If added" procedures trigger when a value is written into a slot (eg. TRAVEL might be written
into the WHAT slot as a result of carrying out the "if needed" and "if required" procedures
45 resulting in instantiation of the TRAVEL slot). "If inconsistent" procedures trigger if the data   45
proposed for the value of a slot is unsatisfactory.
    Any slot preceded by a star (★) is read following instantiation of its frame. A frame with a
starred slot would normally, as in the above example, have an empty value slot and associated
procedures would be triggered, initiating a cycle of events designed to provide a value for the
50 starred slot. As the system acquires knowledge, instances of generic frames are created and   50
their values filled in, so that details relating to the concept represented by the parent generic
frame are supplied.
    In a voice information service, the system must first acquire knowledge about the user's
enquiry and then provide an answer. The relevant frames are instantiated and the relevant values
55 obtained by a sequence of questions being asked of the user, and, once the enquiry has been   55
established, ie when instantiation of the top-level DIALOGUE frame is complete, the database is
accessed. The answer to the user's enquiry is then transmitted.
    The sequence of events following instantiation of the DIALOGUE frame might be as follows:
    1 The starred slot WHAT is needed.
60 2 The "if-needed" procedure 'ask (WHAT)' is triggered resulting in the caller being asked   60
about the type of information required (travel or cost).
    3 The caller responds indicating that he wants travel information, so the value of the WHAT
slot is set to & TRAVEL.
    4 The "check" procedure confirms that 'TRAVEL' is an appropriate value for the WHAT slot.
65 Then the "if added" procedure 'instantiate (WHAT)' is triggered causing an instance of the   65

TRAVEL frame to be created.
The creation of the TRAVEL frame causes further procedures to be activated resulting in more questions to the user until all the needed slots of the top-level starred DIALOGUE frame are filled. At this point the system has acquire sufficient information to be able to answer the query.
5   Examples of five more generic frames are as follows: 5

```
     (i)  Frame      &TRAVEL
          LEAVE      &EVENT        ? instantiate(LEAVE,
10                                      default(LOCATION,Manchester))        10

          ARRIVE     &EVENT
          *TRAIN     &TRAIN        ? lookup(TRAIN,LEAVE,ARRIVE,)
15                                 + tell-user-about(TRAIN)                  15


    (ii)  Frame      &EVENT
20        *LOCATION ------         ? ask(LOCATION)                           20
                                   + confirm(LOCATION)

          *TIME      &INTERVAL
25                                                                           25

   (iii)  Frame      INTERVAL
          SOONEST    &MOMENT
30        *ABOUT     &MOMENT       ? ask(ABOUT)                              30
                                   + compute(SOONEST,ABOUT,LATEST)

          LATEST     &MOMENT
35                                                                           35

    (iv)  Frame      &MOMENT
          HOUR       ------
40        MINUTE     ------                                                  40


     (v)  Frame      &TRAIN
45        FROM       ------                                                  45
          TO         ------
          START      &MOMENT
50        FINISH     &MOMENT                                                 50
```

Frame instantiation would proceed as follows:
1. A TRAVEL frame called TRAVEL is instantiated by the travel slot of the DIALOGUE frame as described above (ie the caller responded to the enquiry initiated by the "if needed" procedure
55 by saying that information on train times was required).   55
2. The star on the TRAIN slot causes the "if-needed" procedure "lookup" to trigger. This is a database access procedure which finds a train from the timetable satisfying the LEAVE and ARRIVE arguments. Before lookup can be applied, the values of the arguments must be known. Thus, the LEAVE and ARRIVE slots must be read in turn.
60 3. The LEAVE slot of TRAIN is read, causing the associated "if-needed" procedure to   60
trigger. This causes the LEAVE frame to be instantiated, creating an instance of the EVENT frame. The value Manchester is placed in the LOCATION slot: in this example, Manchester is the default departure place because the system is based in Manchester and the user is initially assumed to want to travel from there.
65 4. The instantiated EVENT frame has two starred slots. The LOCATION slot has been given a   65

SPECIFICATION

Dialogue system

5 The present invention relates to an interactive dialogue system. Such a system may, for   5
example, operate over the public switched telephone network (PSTN) to provide the telephone
user with a wide range of services and facilities. Services which could be provided include
information services, such as train timetable information; bank balance enquiries; booking facili-
ties for airline, theatre tickets etc; cash transaction services; and control of appliances such as
10 central heating systems, cookers and other household or industrial appliances. Alternatively the   10
system could be used for accessing a computer from a workstation in an office.
   The applicants have developed an interactive voice service. They have conducted trials of a
train timetable information service in which a speech synthesiser is used to ask questions and
the user answers by pressing the appropriate keys on a multi-frequency signalling (MF4) tele-
15 phone. All user responses must be numeric: appropriate questions such as "If you want to   15
travel to Ipswich press 1, if you want to travel to Norwich press 2" are asked by the voice
synthesiser. When sufficient details of the planned journey have been given, a database is
consulted for the times of suitable trains, and these are announced to the caller.
   Such a service would be limited in its application. Callers must have access to a multi-
20 frequency telephone or an acoustically coupled mf sender. In fact most customers have dial   20
telephones, so that much potential revenue would be lost. As indicated above, many answers
are not naturally numeric, and the system can offer only a small number of possible answers
(eg. destinations) for the user to choose between.
   Recorded information services are widely used and it is envisaged that a large market would
25 emerge for interactive information services which are easy to use and can be accessed from any   25
telephone. It is believed that the wide range of services available would create a heavy demand.
   Speech recognisers have been the subject of much research and they are used for a variety of
applications. Output modules such as speech synthesisers are also being widely developed. The
present invention provides an interactive dialogue system incorporating both a speech recogniser
30 and an output module for conducting a dialogue with a user. The system is such that the   30
dialogue with the user can be relatively complex, ie the system exhibits intelligence or semi-
intelligence.
   According to the present invention, there is provided an interactive dialogue system, compris-
ing a speech recogniser arranged to analyse a user's utterances, transmission means for
35 transmitting messages to the user and a dialogue controller including an intelligent knowledge   35
base comprising frame based knowledge representation having a hierarchy of frames containing
information about the dialogue, wherein the dialogue controller is arranged to accept and inter-
pret output relating to a user's utterance from the speech recogniser and to supply data to the
transmission means for the transmission of a message to the user.
40 The system is normally used for responding to a user request, and the dialogue controller is   40
arranged to transmit directions relating to the request to an auxiliary device.
   The auxiliary device may comprise a data store containing data necessary for responding to
the user's request and the dialogue controller is arranged to supply the response data to the
transmission means.
45 Where the user request is to operate a device (eg. an oven, a central heating system or a   45
robot) the directions transmitted by the dialogue controller carry out the required operation.
   The present invention also provides a method of conducting a dialogue with a user to
establish a request, comprising supplying voice signals derived from a user utterance to a
speech recogniser; supplying output from the speech recogniser relating to the voice signals to a
50 dialogue controller including an intelligent knowledge base comprising frame based knowledge   50
representation having a hierarchy of frames containing information about the dialogue; interpret-
ing said output; transmitting a message to the user; and repeatedly interpreting output relating to
user voice signals from the speech recogniser and transmitting messages to the user to establish
the user request, and responding to that request.
55 The present invention will now be described, by way of example, with reference to the   55
accompanying drawings in which:
   Figure 1 is a block diagram of the overall structure of a voice information system according to
an embodiment of the invention;
   Figure 2 is a block diagram showing part of the system of Fig. 1 in greater detail;
60 Figure 3 is a diagram showing a possible simplified frame structure for the dialogue controller   60
of the system of Figs. 1 and 2; and
   Figure 4 is a diagram showing possible phrase structures produced by the linguistic processor
of the dialogue controller of the preceding figures.
   The system shown in Fig. 1 comprises a voice input module 11, a linguistic processor 13, an
65 intelligent knowledge base 15 linked to an information data base 17 and a voice output module   65

19. The voice input module 11 is a speech recogniser such as Logica's "Logos" connected word recogniser and the output module 19 is suitably a speech synthesiser such as Speech Plus "Prose 2000". For a user accessing the system using a telephone, voice input and output are clearly the most convenient; at a computer terminal, output on a VDU may be an alternative.

5 Ideally, a dialogue system should be capable of holding an intelligent dialogue with the user both in clarifying the request and in dealing with any recognition errors. It should also be capable of accommodating more powerful input and output modules without major redesign, and be application independent so that the system can be modified relatively easily for a different application.

10 Typically, an intelligent system should ask only relevant questions; make sensible assumptions and deal within precise answers; use answers which do not follow directly from the question; accept and make use of unsolicited but relevant information (eg. an answer to a question not yet asked); and confirm, preferably, all information supplied by the user.

Currently available speech recognisers have limited vocabularies. Taking this into account and

15 also the need to minimise the complexity of the other components, the system is adapted as described below to exhibit sufficient intelligence to conduct a successful transaction within a limited domain of discourse.

Dialogue is controlled by a dialogue controller which comprises intelligent knowledge base 15 and linguistic processor 13. The intelligent knowledge base 15 incorporated within the dialogue

20 controller comprises a purpose built software process that uses a frame based knowledge representation scheme to encode expertise about dialogue control for the applications task that the system is programmed to perform. The dialogue controller acts as an intermediary between the user and the device provided for the applications task; in this example the device is a data store which stores information necessary for responding to a user enquiry. For other applica-

25 tions, the device may be a domestic or industrial appliance or a computer.

The system operates in response to speech from a user, who answers questions in natural language posed by the system. A microphone converts the user's acoustic signal to an electrical analogue signal and transmits this to the "Logos" speech recogniser 11. Recogniser 11 samples and stores the signal in coded form. The recogniser maintains a store of representations of

30 words in a selected vocabulary and uses these to classify the input in terms of the vocabulary words. The dialogue controller supplies the recogniser with predictions about word order which are made use of in recognising words spoken by the user. The resultant word sequence is transferred to the dialogue controller. The dialogue controller may receive additional information relating to speech recognition such as recognition confidence levels.

35 The dialogue controller performs the principal task of maintaining a dialogue with the user; its processes perform knowledge representation and linguistic functions in order to interface with the speech recogniser and speech synthesiser.

The system shown is adapted to provide a train timetable information service. Database 17 stores train timetable information which is supplied in response to an instruction from the

40 intelligent knowledge base 15. The information is used for a reply to the user via linguistic processor 13 and the speech synthesiser 19.

The components of the system of Figs. 1 and 2 will now be described in greater detail. As indicated above, the dialogue controller is the intelligent part of the system and the intelligent knowledge base system (IKBS) 15 co-ordinates the dialogue.

45 IKBS 15 uses frames to represent knowledge. Frames are a well-established technique in Artificial Intelligence: See, for example Minsky, M., "A Framework for Representing Knowledge", The Psychology of Computer Vision, Ed. Winston, P., McGraw-Hill, New York, 1975. A frame is a package of information about a particular piece of knowledge. Frames are linked together by an inheritance hierarchy, which enables frames representing specific knowledge about a concept

50 to inherit features from higher level frames representing generalisations of that concept. Each frame consists of one or more slots and each slot has a value denoting one particular aspect of the knowledge of that frame. This value may be atomic (eg. a name or number), a reference to a frame lower down in the hierarchy (a sub-frame), or a procedure (often called a "demon"). In existing frame systems, an external agent is required to co-ordinate the sequence of operating

55 software procedures. In this system, the IKBS has knowledge about its own operating behaviour embedded within it, so that the IKBS functions autonomously: the procedures are executed automatically.

In order to achieve autonomy for the IKBS, conventional software routines are used to produce constructs which operate like artificial intelligence constructs.

60 As an example, the following represents a simplified "DIALOGUE" frame: